TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

BAKALÁŘSKÁ PRÁCE

Liberec 2013 **Martin Bumba**

TECHNICKÁ UNIVERZITA V LIBERCI

Fakulta mechatroniky, informatiky a mezioborových studií

Studijní program: B2646 – Informační technologie

Studijní obor: 802R007 – Informační technologie

Návrh webového rozhraní pro práci s databází diplomových a bakalářských prací

A web interface for interacting with the database of master and bachelor's thesis

Bakalářská práce

Autor: **Martin Bumba**

Vedoucí práce: doc. Ing. Josef Chaloupka, Ph.D.

Konzultant: Ing. Karel Paleček

V Liberci 17. 5. 2013

Prohlášení

Byl jsem seznámen s tím, že na mou bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb., o právu autorském, zejména § 60 – školní dílo.

Beru na vědomí, že Technická univerzita v Liberci (TUL) nezasahuje do mých autorských práv užitím mé bakalářské práce pro vnitřní potřebu TUL.

Užiji-li bakalářskou práci nebo poskytnu-li licenci k jejímu využití, jsem si vědom povinnosti informovat o této skutečnosti TUL; v tomto případě má TUL právo ode mne požadovat úhradu nákladů, které vynaložila na vytvoření díla, až do jejich skutečné výše.

Bakalářskou práci jsem vypracoval samostatně s použitím uvedené literatury a na základě konzultací s vedoucím bakalářské práce a konzultantem.

Datum: 17. 5. 2013

Podpis:

Poděkování

V první řadě bych chtěl poděkovat vedoucímu bakalářské práce doc. Ing. Josefu Chaloupkovi, Ph.D. za užitečné konzultace, rady a věcné připomínky k řešení této práce. Dále bych chtěl vřele poděkovat své rodině, přítelkyni a přátelům, kteří mě při vypracování této práce podporovali.

Abstrakt

Bakalářská práce se zabývá vytvořením webového rozhraní umožňujícího fulltextově vyhledávat v kvalifikačních pracích, vytvořených pro Technickou Univerzitu v Liberci. Webové rozhraní je naprogramováno v jazyku PHP a obsahuje dvě oddělené části – veřejně přístupný fulltextový vyhledávač a zabezpečené rozhraní pro správu. Pro uchování všech dat o kvalifikačních pracích je využita databáze MySQL.

Práce dále obsahuje aplikaci určenou k vytváření textových přepisů jednotlivých kvalifikačních prací. Tato aplikace je naprogramována v jazyku Python a pracuje po dávkách určených konfiguračním souborem. Aplikace není umístěna na straně webového rozhraní.

Dokumentace je rozdělena na tři části. A to teoretickou, ve které jsou popsány využité technologie, použitá literatura a porovnání se současnými vyhledávači publikací na internetu. V druhé části je podrobně rozebrán postup a problémy vzniklé při vytváření rozhraní a výše zmíněné aplikace. Dále jsou zde uvedeny příklady a možnosti použití těchto rozhraní. V poslední části je uvedeno celkové zhodnocení bakalářské práce.

Klíčová slova: webové rozhraní, vyhledávání, fulltext, bakalářské práce, diplomové práce, kvalifikační práce

Abstract

This bachelor's thesis deals with creating a web environment allowing full-text search in qualification thesis created for TUL. The web environment is written in PHP and contains two separate parts – a publicly accessible full-text search engine and a secured interface for administration. A MySQL database is used for storing all data.

The work further includes an application designed for creating transcriptions of each project. This application is written in Python and opperates in doses set by a configuration file. The application is not located on the side of the web environment.

Documentation is split into three parts. The first part (theoretical), which contains a description of used technologies and literature and a comparison with current search engines of publications on the internet. The second part, which contains a detailed description of the process of creating said program and problems, that have occured, as well as examples of ways to use such interfaces. And the last part, which contains a general evaluation of the project.

Keywords: web interface, search, fulltext, bachelor's thesis, disertations, qualification thesis

Obsah

Seznam obrázků

Seznam tabulek

Seznam příloh

Seznam použitých zkratek

FTP – File Transfer Protokol

PHP – PHP: Hypertext Preprocessor (rekurzivní zkratka)

SQL – Structured Query Language

GPL – General Public License

XML – Extensible Markup Language

MARC21 – MAchine-Readable Cataloging 21

EXE – Executable (spustitelný)

PDF – Portable Document Format

XHTML – Extensible HyperText Markup Language

CSS – Cascading Style Sheets

Mysqli – MySQL Improved Extension

JPEG – Joint Photographic Experts Group

OCR – Optical Character Recognition

XSS – Cross-site scripting

Úvod

Historie webových vyhledávačů je datována od roku 1990, kdy byl spuštěn první webový vyhledávací nástroj s názvem *Archie*. Tento program umožňoval prohledávat adresáře FTP serverů pomocí regulárních výrazů. *Archie* byl nainstalován na počítači klienta a prohledával pouze předem určené FTP servery.

Dnešní webové vyhledávače mají svou minulost již od roku 1993, kdy byly téměř současně spuštěny tři vyhledávací systémy: *Jumpstation*, *WWW Worm* a *Respository Based Software Engineering (RBSE)*. Tyto vyhledávače rekurzivně procházely internetovou síť webových stránek skrze hypertextové odkazy. *Jumpstaion* a *WWW Worm* při průchodu ukládaly pouze titulky stránek. *RBSE* ukládal malou část zdrojového kódu stránky. V dubnu roku 1994 vznikl první plně fulltextový webový vyhledávač s názvem *WebCrawler*. Ten indexoval nejenom hlavičku stránky, ale také celý obsah (fulltext). Později začaly vznikat další, např. současná *AltaVista* nebo *Google*. Tyto vyhledávače používají velmi sofistikované algoritmy při procházení webových stránek a při vyhledávání pracují velmi intuitivně.

Knihovna Technické univerzity v Liberci (dále jen TUL) v současné době disponuje systémem *PORTACO OPAC 2.0* vyvinutým firmou *KP-SYS*, který umožňuje vyhledávat v kvalifikačních pracích napsaných pro TUL. Tento systém umožňuje vyhledávat pouze v informacích, které jsou ručně vkládány zodpovědnými pracovníky do knihovní databáze. Každý záznam v knihovní databázi může obsahovat: název práce, identifikátor, autora práce, rok vydání, abstrakt práce, fakultu studenta, signaturu, informace o počtu stran a další... Systém *PORTACO OPAC 2.0* má výhodu v poměrně malé databázi a vyšší rychlosti při vyhledávání, ale neumožňuje vyhledávat v textu jednotlivých prací tzv. fulltext. Tato skutečnost snižuje celkovou efektivitu vyhledávacího rozhraní. Pokud je totiž hledaná fráze obsažená pouze v obsahu práce, nelze ji pomocí systému *PORTACO OPAC 2.0* vyhledat. Univerzitní knihovna proto zadala požadavek na vytvoření systému, který bude schopný vyhledávat nejen v obecných informacích, ale také v textu kvalifikačních prací. Nový systém musí fungovat nezávisle na původním systému *PORTACO OPAC 2.0*.

1 Technologie návrhu rozhraní

Práce je z technologického hlediska rozdělena na čtyři části. První část zastupuje úložiště dat ve formě databázové vrstvy. Databáze musí pro správnou funkčnost splňovat následující kritéria – možnost velkých datových sloupců např. typu *mediumtext*, možnost fulltextových indexů a rozhraní pracujícího v jazyku PHP. Tato kritéria od verze 4.1.1 splňuje multiplatformní databáze MySQL, která je volně dostupná pro nekomerční účely pod licencí GPL. MySQL je vyvíjena firmou *Oracle Corporation* a od počátku vývoje byla optimalizována na rychlost, a to i za cenu některých zjednodušení. MySQL umožňuje pracovat ve více vláknech, což umožňuje obsluhu více dotazů najednou. Komunikace s touto databází, jak její název napovídá, probíhá v jazyku SQL. Pro pohodlnou obsluhu databáze je na serveru nainstalována aplikace *phpMyAdmin*, která umožňuje správu objektů v databázi přes webové rozhraní. MySQL server v našem případě beží ve verzi 5.5.29. [1]

V druhé části je řešen problém importu vybraných dat z původní knihovní databáze do nově vytvořené databáze typu MySQL. Data z knihovní databáze byla nejdříve vyexportována ve formátu bibliografických dat MARC21 do souboru typu XML. Dále byl vytvořen PHP skript umístěný na straně serveru, který umožňuje data ze souboru dávkově naimportovat do nově vzniklé databáze MySQL.

Třetí část byla vytvořena pro dávkové generování textových přepisů kvalifikačních prací. Je naprogramována v jazyku Python a využívá nástroj *pdftotext*. Jazyk Python je interpretovaný, což znamená, že pro jeho běh je nutný zdrojový kód programu (skript) a také zvláštní program, který zdrojový kód vykoná (interpretuje). V jazyku Python existuje volně dostupná knihovna *pywin32*, která umožňuje zkompilování, přeložení a následné uložení skriptu jako spustitelnou aplikaci pro operační systém Windows ve formátu EXE. Pro zjednodušení práce s aplikací a zajištění její přenosnosti byla tato knihovna použita. Následně byl vytvořen spustitelný soubor *ctcompilable.exe*. Ke správnému běhu vytvořené aplikace je potřeba operační systém Windows od verze XP a přiložený nástroj *pdftotext*. Vytvořená aplikace není umístěna na straně serveru. Po vytvoření dávky jsou textové přepisy zkomprimovány metodou ZIP, z čehož vyplývá, že výstupem dávky je pouze jediný soubor typu ZIP.

Pdftotext je volně dostupný nástroj z balíčku XPDF [2]. Součástí je převodník PDF souborů na čistý text nebo do jazyku PostScript. Dále obsahuje nástroje určené např. k extrakci obrázků či fontů ze souborů PDF. Celý balíček je naprogramován v jazyku C/C++. Nástroj *pdftotext* pracuje velmi korektně, dobře si poradí i s diakritickými či jinými speciálními znaky. S ohledem na programovací jazyk Python, ve kterém je aplikace pro generování textových přepisů naprogramována, by se mohlo zdát vhodnější využít pro převod PDF souborů na text nějakou volně dostupnou knihovnu jazyku Python např. *PyPdf* nebo *PDFMiner*. Bohužel využití těchto knihoven bylo značně neefektivní. Ani jedna z knihoven nedokázala korektně pracovat s českými diakritickými znaky. Navíc knihovna *PDFMiner* je až 20x pomalejší než nástroj *pdftotext* z balíčku XPDF. Nástroj je obsluhován z příkazové řádky. Po zavolání dokáže zpracovat jeden vstupní soubor a výstup vypsat na klasický výstup *stdout*. Celý balíček XPDF je distribuován pod licencí GPL.

Poslední část zastupuje webové rozhraní umožňující pohodlné vyhledávaní a správu databáze. Rozhraní bylo napsáno v jazyku PHP za využití technologií XHTML, CSS a JavaScript. Hlavní použití jazyka PHP [3] je k programování dynamických webových stránek. Jazyk PHP je interpretovaný, proto je serveru nainstalován jeho interpret ve verzi 5.4.6. Interpret obsahuje mnoho užitečných knihoven a rozšíření. Z nich stojí za zmínku určitě rozšíření *Mysqli*, které v rozhraní zajišťuje komunikaci s databází MySQL. Všechny metody a funkce jsou podrobně popsány v dokumentaci k jazyku PHP, dostupné online na adrese *http://cz2.php.net/manual/en/*. PHP skripty webového rozhraní generují na výstupu kód ve formátu XHTML tak, aby výsledné stránky byly zobrazitelné běžně používanými webovými prohlížeči. Vzhled obou webových rozhraní je nadefinován pomocí kaskádových stylů CSS. XHTML kód i kaskádové styly jsou validní podle konsorcia W3C. JavaScript byl využit ve formě volně dostupné knihovny *jQuery*, rozšíření *jQuery UI* a dále pluginu *Colortip*. JavaScript obsluhuje interaktivní volbu parametrů ve vyhledávacím rozhraní. Plugin *Colortip* je použit pro zobrazení informací při najetí na určité odkazy. Rozšíření *jQuery UI* je využito pro efektní zobrazení nápovědy vyhledávacího rozhraní.

1.1 Porovnání současných vyhledávacích rozhraní

V současné době existuje na internetu mnoho vyhledávacích rozhraní, které umožňují vyhledávat v celosvětové databázi knih, časopisů, veřejných publikacích a dalších zdrojích informací. Bylo tedy vhodné se z hlediska funkčnosti inspirovat právě těmito rozhraními. Pro porovnání rozhraní této bakalářské práce, byla zvolena dvě světově uznávaná vyhledávací rozhraní: Web of Knowledge (*http://webofknowledge.com*) a IEEE Xplore (*http://ieeexplore.ieee.org*), protože mají na internetu již dlouhou historii a obsahují rozsáhlé databáze dat. V tabulce (1) na následující stránce je uvedeno toto srovnání.

Z tabulky je zřejmé, že porovnávaná vyhledávací rozhraní nabízejí spoustu vyhledávacích parametrů. Pracují velice rychle a spolehlivě vzhledem k rozsahu jejich databází. Zásadním rozdílem mezi rozhraním bakalářské práce a dvou vybraných rozhraní je možnost použití českého jazyka pro fulltextové vyhledávaní, což je výhodou pro kvalifikační práce psané pro TUL, které jsou z velké části psané v tomto jazyce.

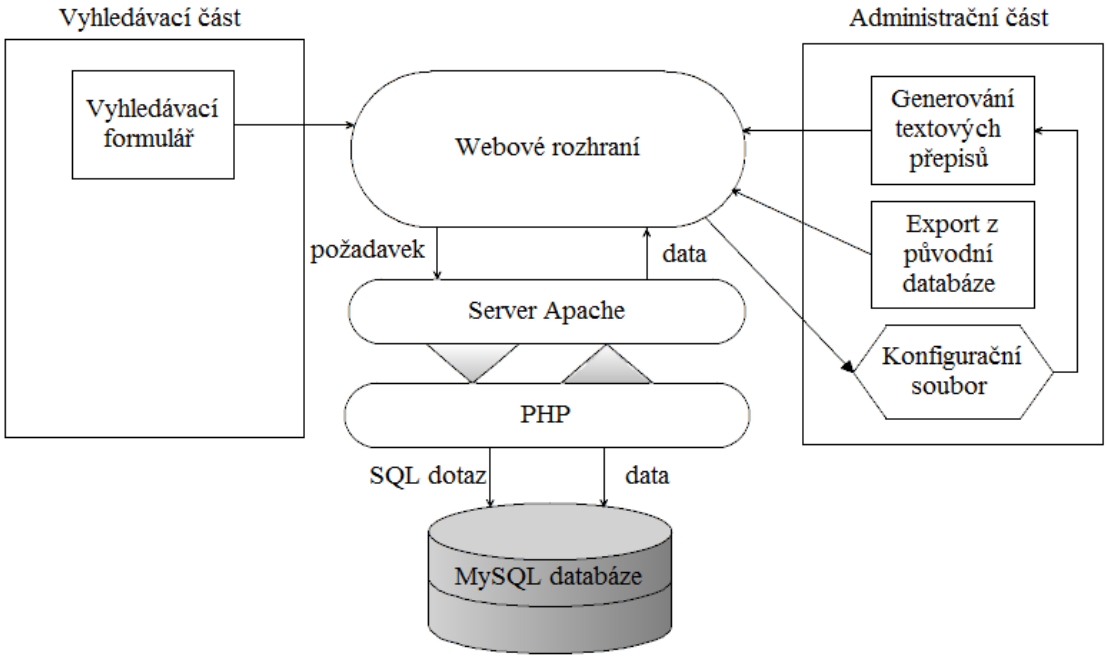
	Web of Knowledge	**IEEE Xplore**	**Rozhraní bakalářské práce**
Veřejně přístupné	ne	ano	ano
Vyhledávací jazyk	angličtina	angličtina	čeština
Volba typu práce	ano	ano	ne, pouze kvalifikační práce
Volba prohledávací databáze	ano	ne	ne, pouze jedna databáze
Možnost prohledávat ve fulltextu	ne	ano	ano
Specifikace ročníku	ano	ano	ano
Možnost specifikovat nakladatele	ne	ano	ne, jeden nakladatel
Přidávání a úprava vyhledávacích parametrů	ano	ano	ano
Volba Booleanovských operátorů	ano	ano	ano
Volba seřazení výsledků	ano	ano	ano
Stránkování výsledků	ano	ano	ne
Zobrazení abstraktu u výsledků	ano	ano	ano
Další možnosti	ano	ano	ano

tabulka 1: srovnání vyhledávacích rozhraní

1.2 Typická architektura webových rozhraní

Celé rozhraní bylo navrženo typickou třívrstvou architekturou obsahující prezenční, aplikační a datovou vrstvu. Díky této architektuře není nutné při úpravě některé vrstvy výrazně přepracovávat vrstvy ostatní.

Celý model rozhraní je znázorněný na obr. 1. Prezenční vrstva zajišťuje komunikaci s uživatelem přes webové rozhraní. Aplikační vrstva obsahuje logiku, která umožňuje oboustrannou komunikaci s databází. V této práci do aplikační vrstvy spadá server Apache a interpret jazyka PHP. Datová vrstva ve formě MySQL databáze obsahuje veškerá data.



obr. 1: diagram třívrstvého rozhraní

2 Návrh databáze

Databáze pojmenovaná ***knihovna*** je hlavní součástí celého rozhraní. Obsahuje všechna data o jednotlivých kvalifikačních pracích a jak již bylo zmíněno v předchozí části je typu MySQL. Databáze musí uchovávat o každé kvalifikační práci následující informace: identifikátor, název, autora, vydavatele, abstrakt, plný text práce, typ kvalifikační práce (diplomová práce, bakalářská práce, ..), fakultu, obsáhlost, rok vydání, signaturu, knihovní identifikátor, předpokládaný název souboru, údaj o poslední modifikaci záznamu v databázi a klíčová slova.

Při návrhu databáze bylo nutné počítat s velkou mohutností sloupce, který obsahuje plný text práce. Po zvážení všech okolností byl pro tento sloupec nakonec zvolen datový typ *mediumtext*. Ten má povolenou délku 0 až 16 777 215 bajtů a nerozlišuje velikost písmen při porovnávání. Místo, které zabírá v paměti se dá vypočítat jako velikost obsahu + 3 bajty pro zaznamenání délky.

Dále bylo nutné vycházet z předpokladu, že každá kvalifikační práce může být specifikována jedním nebo několika klíčovými slovy. Každé klíčové slovo je obsaženo v jedné nebo ve více kvalifikačních pracích. Vzhledem k tomuto předpokladu bylo tedy nutné vytvořit databázi, která bude obsahovat 3 tabulky.

První tabulka reprezentuje jednotlivé kvalifikační práce a má název ***works***. Další tabulky reprezentují klíčová slova a jejich vazbu ke kvalifikačním pracím. Tabulka obsahující klíčová slova má název ***keywords***. A vazební tabulka propojující klíčové slovo s kvalifikační prací se nazývá ***work_keyword***.

Kvůli fulltextovému vyhledávání nad určitými sloupci byl u všech tabulek nastaven typ úložiště MyISAM. Kódování tabulek a jednotlivých sloupců bylo nastaveno na *utf8_czech_ci*. V nastavení serveru MySQL byla oproti výchozím hodnotám upravená pouze hodnota *ft_min_word_length*, která byla nastavena na hodnotu 3. Což znamená, že nejkratší fulltextově vyhledávaná fráze nebo slovo může mít 3 znaky.

2.1 Struktura tabulky *keywords*

Název sloupce	Datový typ	Popis
id	int(11)	Identifikátor (PK)
keyword	varchar(255)	Klíčové slovo

tabulka 2: struktura tabulky *keywords*

2.2 Struktura tabulky *works*

Název sloupce	Datový typ	Popis
id	int(11)	Identifikátor (PK)
name	varchar(255)	Název práce
author	varchar(255)	Příjmení, jméno autora
publisher	varchar(255)	Vydavatel
abstract	text	Abstrakt
text	mediumtext	Plný text práce
type	varchar(100)	Typ
faculty	varchar(3)	Fakulta
pages	varchar(255)	Počet stran
signature	varchar(100)	Signatura
filename	varchar(100)	Předpokládaný název souboru
year	year(4)	Rok vydání
library_id	int(11)	Identifikátor v původní databázi
lastmod	datetime	Čas poslední modifikace záznamu

tabulka 3: struktura tabulky *works*

2.3 Struktura vazební tabulky *work_keyword*

Název sloupce	Typ	Porovnávání
id	int(11)	Identifikátor (PK)
work_id	int(11)	Cizí klíč do tabulky *works*
keyword_id	int(11)	Cizí klíč do tabulky *keywords*

tabulka 4: struktura tabulky *work_keyword*

2.4 Spouště („triggery“) nad tabulkou *works*

V tabulce *works* existuje sloupec *lastmod*, který udržuje u každého záznamu v tabulce informaci o jeho poslední úpravě. Nad tabulkou *works* byly vytvořeny dvě spouště – *works_before_insert* a *works_after_update*, které sloupec *lastmod* nastaví automaticky tak, aby nebylo nutné při každé úpravě nebo vytváření záznamu tento sloupec nastavovat. Spoušť *works_before_insert* se spouští při vytváření záznamu a spoušť *works_after_update* se spouští při jeho úpravě. Každá spoušť nastavuje sloupec *lastmod* u konkrétního záznamu na aktuální datum a čas pomocí MySQL funkce *NOW*.

2.5 Vytvoření databáze

Tvorba databáze a všech jejích součástí proběhla v aplikaci *phpMyAdmin* za pomoci SQL dotazů. Nejdříve byla vytvořena samotná databáze, dále byly vytvořeny jednotlivé tabulky a nakonec byly přidány obě spouště. Po úspěšném vytvoření tabulek bylo nutné správně specifikovat fulltextové indexy nad těmi sloupci, nad kterými je požadováno fulltextové vyhledávání. V tabulce *works* to jsou indexy nad sloupci: *text*, *name*, *abstract*, *author*. V tubulce *keywords* bylo nutné vytvořit pouze jeden fulltextový index, a to nad sloupcem *keyword*.

3 Dávkový import dat z knihovní databáze

Data z knihovní databáze byla nejdříve vyexportována knihovním systémem ve formátu bibliografických dat MARC21 do souboru typu XML. Na prvním řádku souboru je tedy nadefinovaná XML hlavička specifikující verzi a kódování. Dále je v souboru vytvořen kořenový element *collection*, ve kterém jsou v elementech *record* a v dalších vnořených elementech typu *controlfield*, *datafield* a *subfield* definována jednotlivá data ke kvalifikačním pracím. Vnořené elementy *controlfield* a *datafield* obsahují atribut *tag*, který specifikuje typ nesených dat. U vnořených elementů *subfield* je typ dat specifikován atributem *code*. Soubor XML má tedy následující tvar:

```
<?xml version="1.0" encoding="windows-1250"?>
<collection xmlns="http://www.loc.gov/MARC21/slim">
    <record>
      <leader>…</leader>
      <controlfield tag="…">…</controlfield>
      …
      <controlfield tag="…">…</controlfield>

      <datafield tag="…" ind1="…" ind2="…">
        <subfield code="…">…</subfield>
        ...
        <subfield code="…">…</subfield>
      </datafield>
      …
      <datafield tag="…" ind1="..." ind2="…">
        <subfield code="…">…</subfield>
        …
        <subfield code="…">…</subfield>
      </datafield>
    </record>
    …
    [další record]
</collection>
```

3.1 Zpracování (parsování) dat z XML souboru

XML soubor je PHP skriptem nejprve načten pomocí funkce *simplexml_load_file* do proměnné *$xml*. Následně je využit cyklus *foreach*, který projde všechny elementy typu *record*. Dále jsou vybrány a zpracovány vnořené elementy typu *controlfield*, *datafield* a *subfield*. Typ informace, kterou element uchovává je rozlišen pomocí

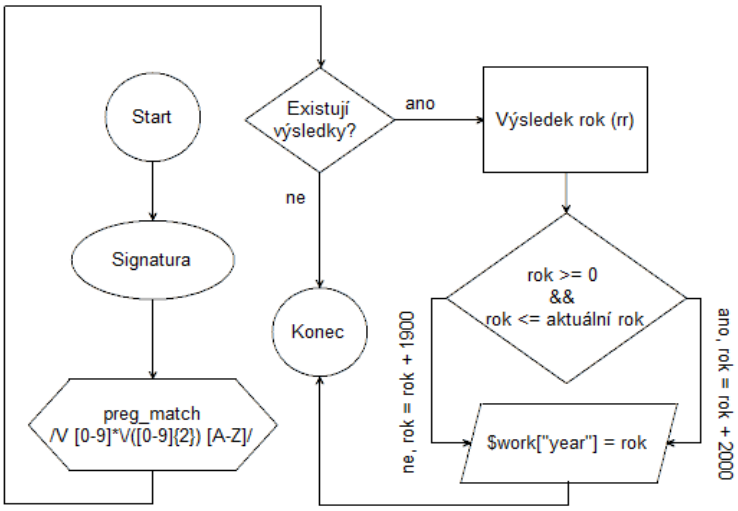
atributů *tag* nebo *code*. Jednotlivé elementy jsou nyní zpracovány a informace o kvalifikačních pracích jsou uloženy do proměnné *$work*, která je typu *array (pole)*.

Složité získávání některých požadovaných informací zpomaluje běh celého importního skriptu. Složitost je dána nekonzistencí některých dat uložených v souboru XML. Identifikátor bakalářské práce je v některých případech uložen jako šestice čísel (např. 297303), jindy je uložen jako řetězec, ze kterého se musí vybrat posledních 6 číslic (např. kpw06460810). Dále stojí za zmínku způsob zjišťování fakulty, pro kterou byla práce vytvořena. Fakulta se dá zjistit pouze ze signatury dané práce. Signatura může mít několik tvarů: *V 35/85 S, V 11/00 Mb, U 723 S*, kde fakultu určuje poslední nebo předposlední znak tohoto řetězce. Pokud je poslední znak B nebo b, jedná se o bakalářskou práci. Výběr fakulty tedy probíhá následovně:

```
$work["faculty"] = "F" . strtoupper(substr(rtrim($sign, "Bb"),
-1));
```

Pomocí funkce *rtrim* je nejprve z pravé strany odstraněn znak B nebo b, pokud existuje. Dále je za využití funkce *substr* vybrán poslední znak, který je následně prostřednictvím funkce *strtoupper* převeden na velký znak. V posledním kroku je na levou stranu připojen znak F označující fakultu. Výsledkem může být např. řetězec *FM* označující fakultu mechatroniky, informatiky a mezioborových studií.

Ročník dané práce je zjišťován přímo z elementu obsahujícího ročník. Lze to, ale pouze v případech, kdy tento element pro danou práci existuje. Pokud neexistuje, je možné ročník práce získat ze signatury následujícím postupem. Nejprve je pomocí regulárního výrazu použitého ve funkci *preg_match* zjištěno, zda signatura obsahuje informaci o ročníku práce. Pokud ano, výsledkem funkce *preg_match* je poslední dvojčíslí ročníku práce. V některých případech dvojčíslí obsahuje úvodní nulu. Proto je důležité každé toto dvojčíslí přetypovat na *integer,* aby se v něm tato nula dále nevyskytovala. Následně je využit ternární operátor, kterým je vyhodnoceno, zdali je získaná informace větší nebo rovna 0 a zároveň menší nebo rovna poslednímu dvoučíslí aktuálního roku. Pokud je podmínka splněna, je přičteno k získané hodnotě 2000. Pokud není splněna je přičteno 1900. Tímto získáme ročník práce ve čtyřčíselném tvaru např. 2000. Postup je zobrazen na obr. 2.



obr. 2: získaní ročníku ze signatury

Klíčová slova jsou získávána přímo z elementů obsahujících tato slova. Pro každou práci jsou zpracována a uložena do pole *$work[“keywords”]*. Po zpracování všech potřebných elementů je záznam kvalifikační práce uložen do databáze, konkrétně do tabulky *works*. Každý ukládaný prvek je ošetřen za pomocí funkce *mysqli_real_escape_string*, čímž je zajištěno vyescapování speciálních SQL znaků. Následujícím SQL dotazem je celý záznam uložen do databáze.

```
INSERT INTO works SET
  `name` = '".@mysqli_real_escape_string($link, $work["name"])."',
  `author` = '".@mysqli_real_escape_string($link,
$work["author"])."',
  `publisher` = '".@mysqli_real_escape_string($link,
union($work["publisher"], " "))."',
  `abstract` = '".@mysqli_real_escape_string($link,
$work["abstract"])."',
  `type` = '".@mysqli_real_escape_string($link, $work["type"])."',
  `faculty` = '".@mysqli_real_escape_string($link,
$work["faculty"])."',
  `pages` = '".@mysqli_real_escape_string($link,
$work["pages"])."',
  `signature` = '".@mysqli_real_escape_string($link,
$work["signature"])."',
  `filename` = '".@mysqli_real_escape_string($link,
$work["filename"])."',
  `year` = '".@mysqli_real_escape_string($link, $work["year"])."',
  `library_id` = '".@mysqli_real_escape_string($link,
$work["id"])."'
```

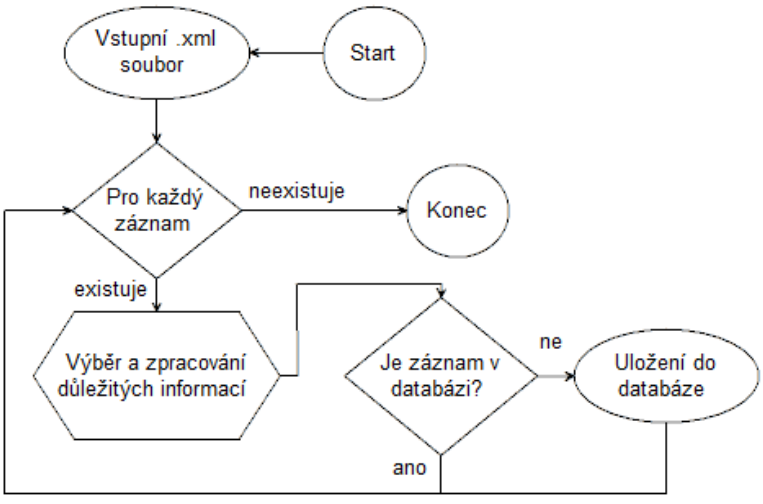
Po vytvoření záznamu v tabulce *works* je zaznamenán jeho identifikátor pomocí funkce *mysqli_insert_id* a uložen do proměnné *$workId*. Nyní jsou pomocí cyklu *foreach* procházena všechna klíčová slova uložená v poli *$work["keywords"]*. Následně je ověřeno, zda se klíčové slovo nachází v tabulce *keywords*. Pokud ano, je do proměnné *$keywordId* zaznamenán jeho identifikátor. V opačném případě je do tabulky *keywords* přidán nový záznam klíčového slova. Vzniklý identifikátor je uložen do proměnné *$keywordId*. Následně je v databázi vytvořen vazební záznam obsahující identifikátor práce (*$workId*) a identifikátor klíčového slova *$keywordId*.

Při provádění importu bylo zjištěno, že v souboru XML některé kvalifikační nemají práce správně zaznamenanou signaturu. Jednalo se o následující problémy:

- signatura nebyla nalezena pod atributem označujícím signaturu
- signatura byla u některých záznamů duplicitní

3.2 Využití dávkového importu

Skript pro dávkový import je implementován v administrační části webového rozhraní v sekci „Importovat z .xml“. Celý postup provedení dávkového importu je zobrazen na obr. 3.



obr. 3: dávkový import dat z .xml souboru

4 Vytvoření textových přepisů

Zásadní problematikou bylo vytvoření textových přepisů k jednotlivým kvalifikačním pracím. Všechny kvalifikační práce jsou v knihovně uloženy ve formátu *PDF* pod cestou: *fakulta/ročník/signatura_práce.pdf* (např. *Fakulta mechatroniky/2011/V 008-11 Mb.pdf*). Pojmem textový přepis je myšlen čistý text práce bez formátování a obrázků. V takto vytvořeném přepisu je možno fulltextově vyhledávat. Textový přepis obsahuje jen důležitou textovou informaci a nezabírá tedy zbytečné místo v paměti. Největší problém představují ty kvalifikační práce, které nejsou uloženy klasickým textovým způsobem, ale jsou vytvořeny obrázkovým tipem PDF souboru. To znamená, že každá strana v PDF souboru je celostránkový sken s určitou kompresí např. JPEG. Tyto kvalifikační práce tedy nejde převést na text klasickým výběrem textu, ale k jejich převedení na čistý text je zapotřebí technologie OCR. Tento způsob převodu nebyl vzhledem k aplikační složitosti součástí této bakalářské práce.

4.1 Převod textového PDF souboru na čistý text

Cílem bylo naprogramovat aplikaci, která bude PDF soubory převádět po dávkách určených vstupním konfiguračním souborem, kde na každém řádku souboru bude relativní cesta k PDF souboru kvalifikační práce. Tento vstupní soubor bude generován v administrační části webového rozhraní. Aplikace bude pracovat v příkazové řádce a budou jí předány dva povinné parametry:

- vstupní konfigurační soubor
- absolutní cesta ke kořenovému adresáři kvalifikačních prací

Dále můžou být aplikaci předány dva nepovinné parametry:

- *f, from* – od jakého PDF souboru začít s převodem
- *c, count* – kolik souborů převést

Při využití aplikace pro vytváření textových přepisů naprogramované v jazyce Python, je načten každý řádek konfiguračního souboru a uložen do proměnné *LIST* struktury typu *list (pole)*. Celý takto vytvořený *list* se dále seřadí, aby později nedocházelo ke zpomalovaní důsledkem nadměrného vyhledávaní souborů na disku.

V dalším kroku je vytvořen dočasný adresář *temp*, do kterého jsou vytvořené přepisy ukládány. Dále je vytvořen *list NOT_READABLE*, který obsahuje relativní cesty PDF souborů, které nepůjdou z nějakého důvodu převést.

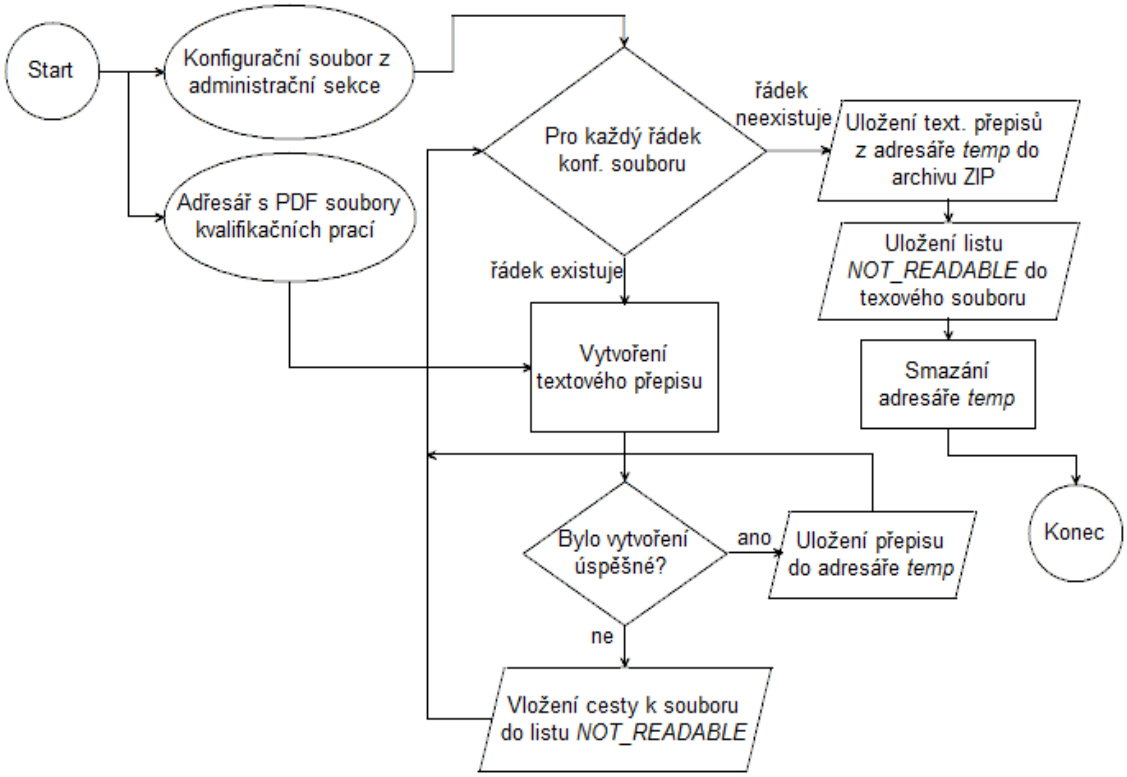
V případě, že nebyly předány nepovinné parametry *-f* nebo *-c,* je projit každý řádek v proměnné *LIST*. V opačném případě jsou projity pouze řádky proměnné *LIST* s ohledem na tyto parametry.

Pomocí knihovny *subprocess* jazyka Python a funkce *check_output* je zavolán nástroj *pdftotext* z volně dostupného balíčku XPDF, který se postará o samotný převod. Funkce *check_output* přečte klasický výstup příkazové řádky po vykonání příkazu. Celý tento proces je popsán následující funkcí:

```
def convert_pdf_xpdf(path):
    try:
        output = subprocess.check_output("\"pdftotext.exe\" -q
-enc \"UTF-8\" \"%s\" -" % path, shell=True)  #zavola externi
nastroj

    except:
        return "" #vyjimka=chyba, vrati prazdny retezec
        raise

    else:
        return ' '.join(output.split())  #smaze zbytecne mezery a
vrati obsah PDF souboru
```

Nejdříve je tedy ověřena existence PDF souboru a jeho velikost. Je důležité, aby soubor byl menší než 30MB. V případě splnění těchto podmínek se skript pokusí provést převod pomocí výše zmíněné funkce *convert_pdf_xpdf*. Pokud funkce vrátí text, jehož délka je vetší než 100 znaků, je tento text uložen do nového textového souboru v dočasném adresáři *temp*. Při nesplnění některé z těchto podmínek je absolutní cesta k souboru uložena do proměnné *NOT_READABLE*. Nakonec je celý obsah adresáře *temp* pomocí knihovny *shutil* a funkce *make_archive* zkomprimován metodou ZIP a uložen do souboru s vygenerovaným názvem za pomoci aktuálního data a času. Po dokončení komprimace je dočasný adresář *temp* smazán. V posledním kroku je vytvořen textový soubor z proměnné *NOT_READABLE,* obsahující absolutní cesty k souborům, u kterých byl převod neúspěšný.

Aplikace byla po naprogramování přeložena a zkompilována za pomocí knihovny *pywin32*. Výsledek kompilace byl uložen jako spustitelný soubor pro Windows od verze XP s názvem *ctcompilable.exe*. V adresáři, ve kterém se výsledná aplikace nachází, musí být umístěn i nástroj *pdftotext*. Bez tohoto nástroje bude aplikace nefunkční. Nástroj se nachází v souboru *pdftotext.exe*. Celý postup generování textových přepisů je znázorněn na obr. 4.



obr. 4: vytvoření textových přepisů

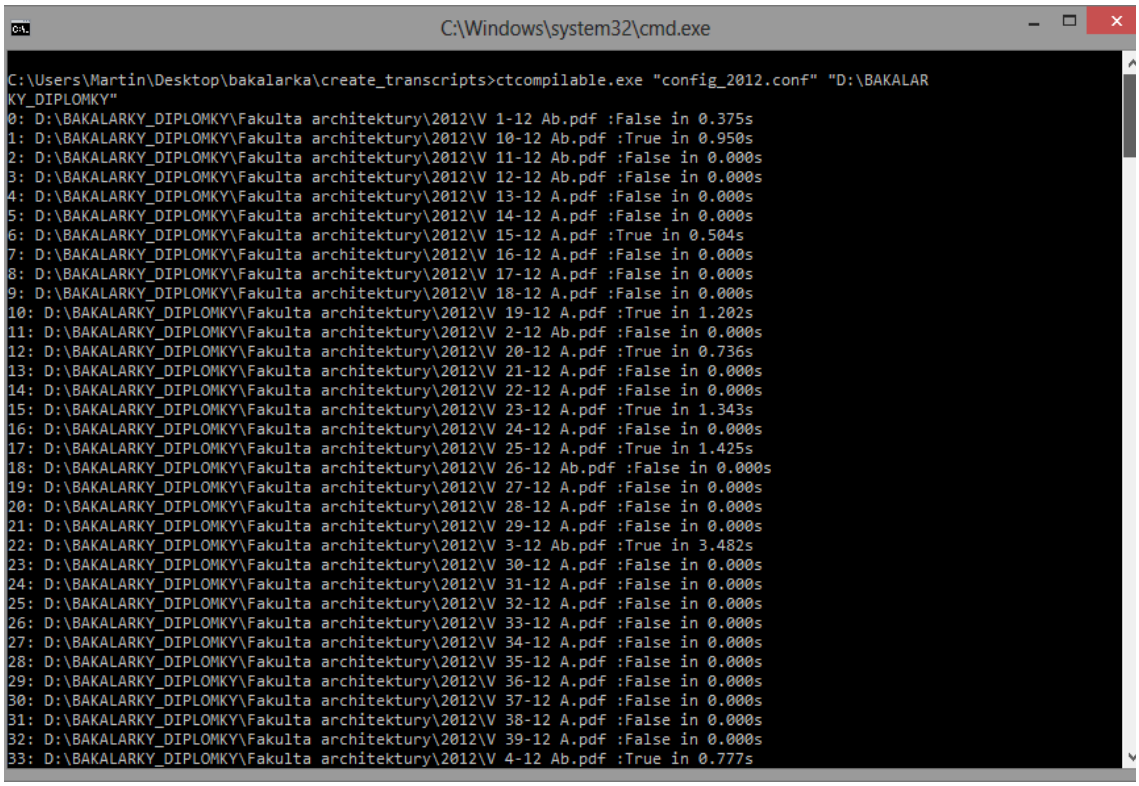
4.2 Standardní využití aplikace

Pro využití aplikace je potřeba mít přímý přístup ke kořenovému adresáři kvalifikačních prací a stažený příslušný konfigurační soubor, pro který mají být textové přepisy vytvořeny. V místě, kde je aplikace umístěna, je spuštěna příkazová řádka a zadán příkaz:

```
ctcompilable.exe "input_config" "input_path"
```

Místo *input_config* je zadána cesta ke konfiguračnímu souboru a za *input_path* je dosazena absolutní cesta ke kořenovému adresáři kvalifikačních prací. Skript bude nyní vypisovat průběžné výsledky pro jednotlivé textové přepisy (obr. 5). Na závěr je

vytvořen soubor obsahující dávku textových přepisů s názvem *output_rrrr-mm-dd_hh-mm.zip,* a zároveň soubor s názvem *not_readable_rrrr-mm-dd_hh-mm.txt* obsahující absolutní cesty k souborům, ke kterým nebylo z nějakého důvodu možné vytvořit textové přepisy. Za *rrrr-mm-dd_hh-mm* bude dosazeno aktuální datum a čas.



obr. 5: průběh tvorby textových přepisů

Vytvořenou dávku textových přepisů je nyní možné naimportovat do databáze v administrační části webového rozhraní, konkrétně v sekci „Importovat textové přepisy“.

5 Webové rozhraní

Webové rozhraní obsahuje dvě samostatné části – administrační a vyhledávací. Administrační část je zabezpečena pomocí přihlašovacího formuláře a umožňuje pohodlnou správu databáze. Veřejně přístupná vyhledávací část je striktně zaměřena na vyhledávání kvalifikačních prací podle zadaných kritérií. Obě části jsou naprogramovány v jazyku PHP za využití technologií XHTML, CSS a JavaScript. Všechny části jsou uložené v kódování *utf-8*.

Pro správnou funkci obou částí je nutný adresář *includes*, umístěný v kořenovém adresáři webového rozhraní. Adresář obsahuje soubory *config.php*, *functions.php*, *driver.php* a skrytý soubor *.htaccess*. Soubor *config.php* je konfigurační soubor webového rozhraní. Nachází se v něm nastavení přístupu do databáze, nastavení přihlašovacích údajů do administrační části a další interní nastavení webového rozhraní. Soubor *functions.php* obsahuje důležité předem nadefinované funkce. V souboru *driver.php* je umístěna řídící logika vyhledávacího rozhraní. Ve skrytém souboru *.htaccess*, který slouží k dodatečné konfiguraci webového serveru je příkaz zabraňující přístupu do adresáře *includes* z venku. Obě rozhraní pracují s databází, proto bylo nutné všechny SQL dotazy na databázi ošetřovat proti útokům *SQL injection*. Každá proměnná předávaná SQL dotazu je nejdříve ošetřena funkcí *mysqli_real_escape_string*, která vyescapuje všechny speciální znaky jazyka SQL a zamezí tím zneužití dotazu.

5.1 Administrační část

Administrační část se nachází v podadresáři *admin* webového rozhraní. Přístup do administrační části je tedy z adresy *http://relator.ite.tul.cz/admin/*.

PHP skript využívá tzv. s*essions* pro udržení informací o přihlášené osobě po dobu spuštění relace nebo do odhlášení. *Sessions* uchovávají na straně serveru veškerá data a na straně klienta pouze jejich identifikátor *phpsessid*. Jsou tedy bezpečnější než jiné metody např. *cookies* nebo *get,* které uchovávají veškeré informace na straně klienta.

Patřičná sekce je zvolena po přihlášení v levém panelu administrační části a je předávána metodou *get* pomocí proměnné *page*. Je tedy obsažena v internetové adrese za otazníkem. Pokud proměnná *page* není definována, zobrazí se úvodní strana administrační části.

Skript nejdříve ověří zdali v *sessions* existují informace o přihlášené osobě. Pokud ano, je zobrazena příslušná sekce administračního rozhraní podle proměnné *page*. V opačném případě je zobrazen přihlašovací formulář (obr. 6). Pokud není uživatel přihlášen, jsou všechna vstupní data ošetřována proti útokům XSS.

obr. 6: přihlašovací formulář

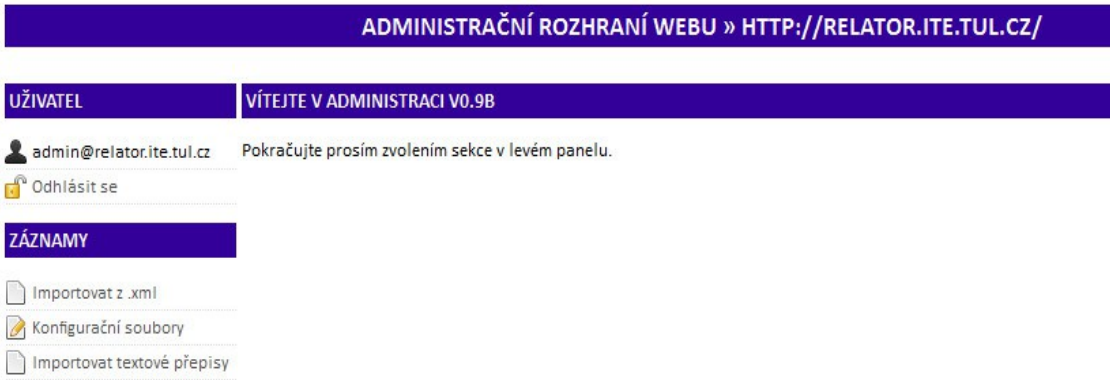
5.1.1 Ošetření proti útokům XSS

Ošetření probíhá pomocí funkce *fixSuperglobals*, v níž jsou rekurzivně procházena všechna pole vstupních superglobálních proměnných *$_GET, $_POST, $_COOKIE, $_REQUEST*. Všechny prvky i klíče jsou ošetřeny funkcí *htmlspecialchars*, která speciální znaky HTML převádí na entity.

5.1.2 Přihlášení do rozhraní správy

Pří zadání přihlašovacích údajů a odeslaní formuláře je za pomoci podmínek ověřena jejich shoda s údaji uloženými v konfiguračním souboru *config.php*. Pro vyšší bezpečnost je heslo v konfiguračním souboru zašifrováno metodou *sha-1*.

Heslo odeslané z formuláře je tedy nutné nejprve zašifrovat funkcí *sha1* a teprve poté porovnat s otiskem uloženým v konfiguračním souboru. Při shodě přihlašovacích údajů jsou nastavena patřičná *sessions* a je zobrazena úvodní stránka administrace (obr. 7). Při zadání neplatných údajů bude uživateli zobrazena chyba o neplatném přihlášení.



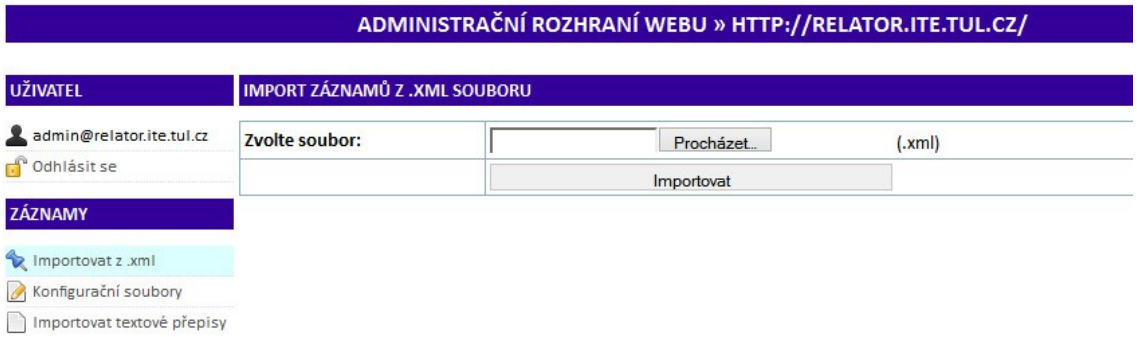
obr. 7: úvodní stránka administrace

Správné přihlašovací údaje zná pan doc. Ing. Josef Chaloupka, Ph.D. V budoucnu budou tyto údaje předány do knihovny TUL, kde bude nově vzniklé rozhraní dále spravováno a rozšiřováno o nové záznamy kvalifikačních prací.

5.1.3 Administrační část – „Importovat z .xml“

V této sekci je možné importovat nové záznamy kvalifikačních prací do databáze. Ve formuláři je zvolen soubor ve formátu XML, ze kterého chceme záznamy naimportovat.

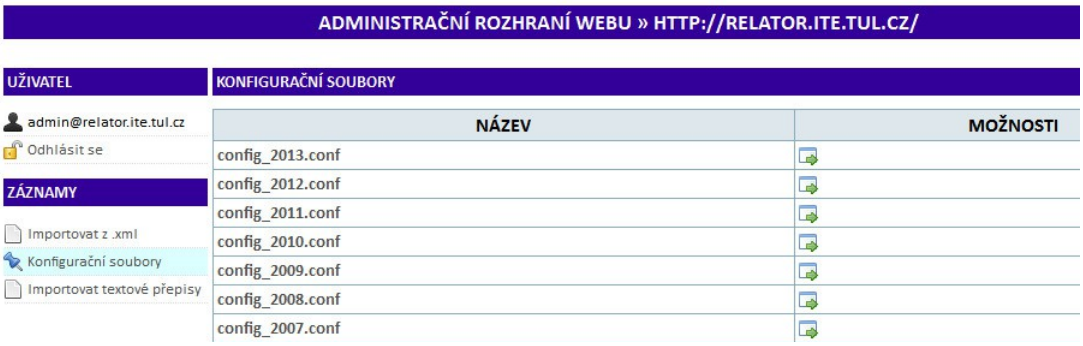
Po stisku tlačítka importovat jsou inkrementační metodou nahrány jen ty záznamy, které databáze ještě neobsahuje. V databázi tedy nedochází k vytváření duplicitních záznamů. Jak skript importu pracuje, je vysvětleno v kapitole 3 . Při úspěšném importu je vypsáno hlášení s počtem nově vytvořených záznamů v databázi. Vzhled sekce je zobrazen na obr. 8.



obr. 8: administrační část – „Importovat z .xml“

5.1.4 Administrační část – „Konfigurační soubory“

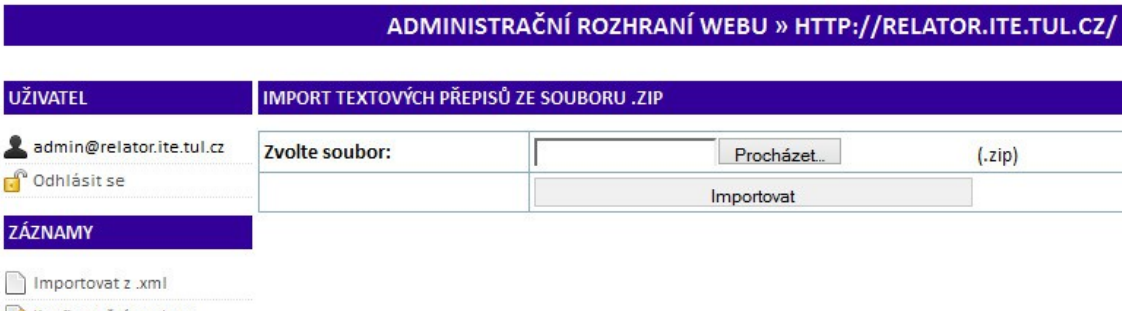
V sekci je možné stáhnout konfigurační soubory, které jsou vygenerovány po ročnících. Každý konfigurační soubor obsahuje relativní cesty k PDF souborům všech kvalifikačních prací za daný rok. Stažený konfigurační soubor je možno předat aplikaci určené k tvorbě textových přepisů a vytvořit tak dávku textových přepisů. Funkce této aplikace je vysvětlena v kapitole 4 . Sekce je zobrazena na obr. 9.



obr. 9: administrační část – „Konfigurační soubory“

5.1.5 Administrační část – „Importovat textové přepisy“

Sekce je zaměřena na import textových přepisů kvalifikačních prací do databáze. Po odeslaní formuláře s vyplněnou cestou k dávkovému souboru typu ZIP je soubor nahrán na server a následně dočasně rozbalen za pomoci třídy *ZipArchive* do pomocného adresáře *temp* umístěného v adresáři *admin*. Následně je uživatel přesměrován do rozhraní importu, který zajišťuje skript *import.php* umístěný také v adresáři *admin*. Skript projde všechny rozbalené textové přepisy a následně je nahraje do databáze ke kvalifikačním pracím. Jako identifikátor je využita *signatura*. Skript v průběhu pro každý textový přepis vypíše hlášeni o úspěchu či neúspěchu uložení přepisů do databáze. Po vykonání skriptu je obsah adresáře *temp* smazán a uživatel přesměrován zpět do administračního rozhraní. Rozhraní pro import je také zabezpečeno pomocí *sessions* proti neoprávněnému použití. Vzhled sekce je zobrazen na obr. 10.



obr. 10: administrační část – „Importovat textové přepisy"

5.1.6 Odhlášení z rozhraní správy

Po stisknutí tlačítka „Odhlásit se“ skript odstraní veškerá vytvořená *sessions* a znovu zobrazí přihlašovací formulář.

5.2 Vyhledávací rozhraní

Vyhledávací rozhraní je nejdůležitější součástí této bakalářské práce. Umožňuje pomocí přesně specifikovaných kritérii fulltextově prohledávat databázi kvalifikačních prací. Je veřejně dostupné na webové adrese *http://relator.ite.tul.cz/*.

Hlavní obsluhující skript *index.php* se nachází v kořenovém adresáři webového rozhraní. Skript nejdříve načte soubory *config.php* a *functions.php* z adresáře *includes*, vytvoří připojení do databáze a ošetří všechny vstupní proměnné proti napadení XSS (popsáno v kapitole 5.1.1). Dále načte soubor *includes/driver.php*, který obsahuje logiku obsluhující vyhledávací formulář.

5.2.1 Obsluha vyhledávacího formuláře

Nejprve je ověřeno odeslání vyhledávacího formuláře pomocí podmínky a funkce *empty*. Poté jsou zkontrolovány vyhledávací parametry a omezení, zdali jsou správně vyplněny. Pří nedodržení některé podmínky při kontrole je do pole *$errors* přidán řádek s vysvětlením chyby. Tyto chyby jsou uživateli vypsány přímo ve vyhledávacím formuláři.

5.2.2 Kontrola omezení na ročník

U obou vstupních polí pro ročník je ověřeno, zdali je zadáno 4znakové číslo. Pokud ano, je u pole *ročník-od* ověřeno, zdali je toto číslo větší nebo rovno minimálnímu ročníku nastavenému v konfiguračním souboru. Pokud podmínky nejsou

dodrženy, je do pole *$errors* uložena následující chyba: „Pole pro *ročník-od* bylo uživatelsky modifikováno!”. U pole *ročník-do* je kontrolováno, zdali je číslo větší nebo rovno než minimální ročník a zároveň menší nebo rovno maximálnímu ročníku nastavenému v konfiguračním souboru. Při nedodržení je do pole *$errors* uložena chyba: „Pole pro ročník-do bylo uživatelsky modifikováno!”. Dále je kontrolováno, zdali není číslo v *ročníku-do* větší než číslo v *ročníku-od*. Pokud ano, je uložena chyba: „Pole pro ročník-do musí být větší nebo rovno než ročník-od!”.

5.2.3 Ověření fakulty a možnosti seřazení výsledků

Pomocí funkce *array_key_exists* se ověřuje, zdali se hodnota z pole pro omezení na fakultu nachazí v klíčích pole fakult *$allowedFaculties.* To je nadefinováno v konfiguračním souboru. Pokud podmínka není dodržena, je uloženo chybové hlášení: „Pole pro výběr fakulty bylo uživatelsky modifikováno!”. Hodnota z pole pro řazení výsledků je kontrolována identicky jako pole pro výběr fakulty s rozdílem, že hodnota z pole je porovnávána s polem povolených možností seřazení *$allowedOrders*. Při nesplnění podmínky je opět uloženo chybové hlášení: „Pole pro způsob řazení bylo uživatelsky modifikováno!”.

5.2.4 Ověření vyhledávacích polí

V dalším kroku jsou zkontrolovány jednotlivé rešeršní řádky pro fulltextové vyhledávání. Každý řádek obsahuje: vstupní textové pole, pole pro Booleanovský výraz (u prvního rešeršního řádku je skryté) a pole pro výběr sloupce. Pole pro Booleanovský výraz a výběr sloupce jsou vybírána z rozbalovací nabídky. Pokud byla modifikována, je přidáno odpovídající chybové hlášení do pole *$errors*. Dále je ověřeno, zda každá vstupní fráze nebo slovo z textového pole má alespoň 3 znaky (minimální délka slova/fráze pro fulltextové vyhledávaní, nastavená na MySQL serveru). Jsou-li tyto ověřující podmínky splněny, je do nadefinovaného pole *$whereStatement* přidán řádek obsahující další pole, kde na indexu 0 je Booleanovský výraz, a na indexu 1 část SQL výrazu pro fulltextové vyhledávání nad určeným sloupcem.

Následně je sestaven kompletní vyhledávací SQL dotaz. SQL dotaz je omezen na výběr pouze prvních 50 nalezených záznamů. Tyto záznamy jsou na stránce přehledně zobrazeny přímo pod vyhledávacím formulářem. V případě, že nebyl nalezen žádný

záznam kvalifikační práce, je vypsáno hlášení: „Bohužel nebyl nalezen žádný výsledek.“. Následuje konkrétní příklad sestaveného SQL dotazu:

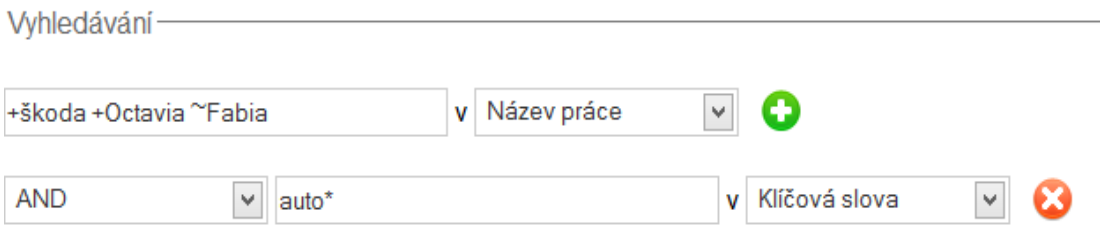
```
SELECT `works`.*
FROM `works`
LEFT JOIN (`keywords`, `work_keyword`)
  ON (`keywords`.`id` = `work_keyword`.`keyword_id` AND
`work_keyword`.`work_id` = `works`.`id`)
WHERE
  MATCH(`text`) AGAINST ('+škoda +\"zadní náprava\" +(fabia
octavia)' IN BOOLEAN MODE)
  AND MATCH(`keyword`) AGAINST ('+plast*' IN BOOLEAN MODE)
  AND `year` >= '1958'
  AND `year` <= '2013'
  AND `faculty` = 'fs'

GROUP BY `works`.`id`

ORDER BY MATCH(`text`) AGAINST ('+škoda +\"zadní náprava\" +(fabia
octavia)' IN BOOLEAN MODE) + 3 * MATCH(`keyword`) AGAINST
('+plast*' IN BOOLEAN MODE) DESC

LIMIT 0, 50
```

5.2.5 Vzhled vyhledávacícho rozhraní

Z hlediska vzhledu obsahuje vyhledávací rozhraní dvě části viz příloha 1. V první (vrchní) části je umístěn vyhledávací formulář, kde je nalevo možné specifikovat vyhledávací kritéria a napravo nastavit další možnosti nebo omezení při vyhledávání. V části pod formulářem jsou vypisovány jednotlivé výsledky vyhledávaní. Pro každý výsledek je zobrazen název kvalifikační práce, abstrakt, klíčová slova a důležité informace: jméno autora, rok vydání, typ kvalifikační práce, signatura, knihovní identifikátor, obsáhlost práce a nakladatele. Abstrakt je zobrazen pouze v případě, že před odesláním formuláře nebyla nastavena možnost nezobrazení abstraktu. Pokud u dané práce existují klíčová slova, jsou vypsána v řádku pod abstraktem práce. Identifikátor kvalifikační práce obsahuje hypertextový odkaz směřující na příslušnou stránku kvalifikační práce na webových stránkách knihovny TUL. Informace o obsáhlosti a nakladateli jsou zobrazeny při najetí kurzorem myši na dané odkazy (*obsáhlost, nakladatel*).

5.2.6 Nastavení vyhledávacích kritérií

Vyhledávací kritéria volíme v levé části formuláře (obr. 11). Pomocí tlačítka *plus* nebo *křížek* je možno přidávat či odebírat jednotlivé řádky rešerše. Každý řádek obsahuje textové pole a rozbalovací nabídku. Textové pole umožňuje přesně specifikovat hledaná slova nebo fráze. Rozbalovací nabídka určuje nad jakým sloupcem bude fulltextové vyhledávání probíhat. Povolené sloupce jsou: název, autor, abstrakt, text práce a klíčová slova. Rešeršní řádky kromě prvního obsahují ještě druhou rozbalovací nabídku. Pomocí ní je zvoleno, jestli rešeršní řádek bude v SQL dotazu připojován k předchozímu přes logické *AND* nebo logické *OR*.



obr. 11: nastavení vyhledávacích kritérií

5.2.7 Specifikace hledaných slov nebo frází v textovém poli

Hledané fráze nebo slova (dále jen výrazy) jsou zadávány do textového pole. Fulltextový vyhledávač pracuje v takzvaném Boolean módu [4], což znamená, že můžeme u hledaných výrazů využívat chytrých operátorů, které jsou obsluhujícím skriptem povoleny. Tyto operátory umožňují upřesnění vyhledávacích požadavků u jednotlivých výrazů. Větší množství výrazů se od sebe odděluje mezerou. Pokud je zadán pouze jediný výraz bez jakéhokoliv operátoru, musí být v hledaném textu obsažen. Při zadání více výrazů bez operátorů musí být v hledaném textu obsažen alespoň jeden z nich. V tabulce 5 jsou zobrazeny všechny skriptem povolené vyhledávací operátory včetně příkladů jejich použití.

Operátor	Umístění	Funkce	Příklad
" " (horní uvozovky)	před a za hledaným výrazem	specifikace přesné hledané fráze	"hledaná fráze"
+	před výrazem	výraz musí být obsažen v hledaném textu	+škoda
-	před výrazem	výraz nesmí být obsažen v hledaném textu	-škoda
~	před výrazem	pokud je výraz obsažen v hledaném textu, zobrazí se ve výsledcích seřazených podle relevance níže, než pokud tento výraz v textu obsažen není (má nižší prioritu)	~škoda
( )	před a za podskupinou výrazů	určení podskupiny výrazů	+(Fabia Octavia) v hledaném textu musí být obsaženo slovo *Fabia* nebo *Octavia*
*	za částí výrazu	operátor nahrazuje část hledaného výrazu	+oct* v hledaném textu musí být obsaženo slovo začínající na *oct*
žádný operátor		výraz nemusí být v textu obsažen, pokud je obsažen, bude ve výsledcích zobrazen výše (má vyšší prioritu)	škoda

tabulka 5: povolené vyhledávací operátory

Použití některých vyhledávacích operátorů je zobrazeno na obr. 12. V případě, že budou zadána kritéria z tohoto obrázku, budou nalezeny kvalifikační práce, které v textu obsahují slovo *škoda*, frázi *"zadní náprava"* a neobsahují slovo *fabia* nebo *octavia*.

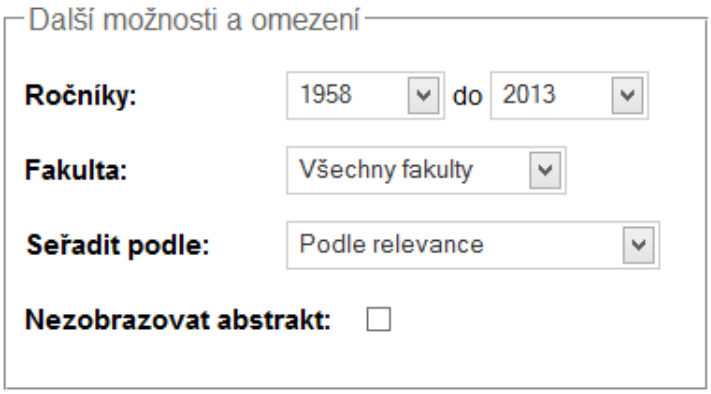
obr. 12: konkrétní specifikace hledaných slov

5.2.8 Zobrazení nápovědy

Po klepnutí na odkaz: „Zobrazit nápovědu", umístěného v pravém dolním rohu levé části formuláře, je možno zobrazit ve vyhledávacím rozhraní tabulku povolených vyhledávacích operátorů (tabulka 5) a další nápovědu. Nápověda bude poté přehledně zobrazena pomocí JavaScriptu a rozšíření *jQuery UI* v plovoucím okně.

5.2.9 Další možnosti a omezení

Pravá část formuláře umožňuje (obr. 13) specifikovat další možnosti a omezení, které není možné specifikovat v levé části formuláře. Je zde možné nastavit **interval ročníků** vyhledávaných kvalifikačních prací. Ve výchozím stavu je interval nastaven od roku 1958 (od tohoto roku jsou datovány první kvalifikační práce na TUL) do aktuálního roku. Toto omezení je voleno pomocí dvou rozbalovacích nabídek. Další možností pravé části je **omezení na fakultu**, pro kterou byla daná práce zhotovena. Výběr probíhá opět pomocí rozbalovací nabídky. Je možno zvolit konkrétní fakultu TUL, nebo výchozí hodnotu – všechny fakulty. Další možností vyhledávacího rozhraní je nastavení **metody seřazení** nalezených výsledků. Ve výchozím stavu je nastaveno řazení podle relevance výsledků. Dále je možné řadit vzestupně i sestupně podle autora, ročníku nebo názvu kvalifikační práce. Volba probíhá opět pomocí rozbalovací nabídky. Poslední možností vyhledávacího rozhraní je **nezobrazování abstraktu** u nalezených kvalifikačních prací. Tato volba je nastavena pomocí zaškrtávacího políčka.



obr. 13: další možnosti a omezení

6 Závěr

V rámci této bakalářské práce bylo vytvořeno webové rozhraní umožňující fulltextově vyhledávat v databázi kvalifikačních prací TUL. Tato databáze obsahuje **29 706** záznamů kvalifikačních prací, z nichž je možno fulltextově vyhledávat v textu v **5 734** pracích. U nich bylo možné vytvořit textové přepisy metodou popsanou v kapitole 4 . Další přepisy ke starším pracím budou vytvářeny metodou OCR. Databáze dále obsahuje 20 979 klíčových slov a 86 967 vazebních záznamů klíčových slov ke kvalifikačním pracím. Databáze má velikost přes **1,1 GB**.

Dále bylo vytvořeno zabezpečené webové rozhraní umožňující tuto databázi spravovat. Je tedy možné přidávat dávkově nové záznamy kvalifikačních prací a také nahrát k těmto záznamům dávkově textové přepisy. Tyto přepisy jsou generovány pomocí aplikace, která byla také vytvořena v rámci této bakalářské práce.

Při vypracování této práce byly nalezeny problémy při importu nových kvalifikačních prací z původní knihovní databáze. Byly zjištěny nekonzisteně vyplněné některé elementy v exportním XML souboru z knihovního systému. Dalším zjištěným problémem byl výskyt duplicitních záznamů v tomto souboru. Tato problematika byla řešena s Mgr. Martou Zizienovou, na základě konzultace došlo k nápravě těchto chyb v knihovním systému. Při testování celého rozhraní bylo zjištěno, že server, na kterém webové rozhraní běží, není dostatečně výkonný pro jeho rychlý a dlouhodobý běh. V následující době bude tedy celé rozhraní přesunuto na výkonnější server. Veškerá administrativa vzniklého webového rozhraní bude předána do knihovny TUL, která bude dále zajišťovat správu databáze kvalifikačních prací a textových přepisů.

Vyhledávací rozhraní v současné době slouží ke snadnému nalezení kvalifikačních prací napsaných pro TUL. Jedno z možných využití do budoucna je použít vzniklou databázi textových přepisů pro vyhledávání plagiátů kvalifikačních prací např. na základě statistiky slov.

Seznam použité literatury

[1] VRÁNA, Jakub. Fulltextové vyhledávání v MySQL. [online]. [cit. 2013-05-02]. Dostupné z: http://php.vrana.cz/fulltextove-vyhledavani-v-mysql.php

[2] GLYPH & COG, LLC. *Xpdf: Home* [online]. [cit. 2013-05-05]. Dostupné z: http://www.foolabs.com/xpdf/home.html

[2] GILMORE, Jason W. *Velká kniha PHP a MySQL 5: kompendium znalostí pro začátečníky i profesionály*. Vyd. 1. [i.e. 2. vyd.]. Brno: Zoner Press, 2007, 864 s. ISBN 80-868-1553-6.

[3] KOFLER, Michael. *Mistrovství v MySQL 5*. Vyd. 1. Překlad Jan Svoboda, Ondřej Baše, Jaroslav Černý. Brno: Computer Press, 2007, 805 s. ISBN 978-80-251-1502-2.

Přílohy

příloha 1: vzhled vyhledávacího webového rozhraní

Obsah přiloženého CD

- Text bakalářské práce ve formátu PDF
- Kompletní webové rozhraní včetně všech zdrojových kódů
- Kód pro vytvoření struktury databáze v jazyku SQL
- Zdrojový kód aplikace pro vytváření textových přepisů v jazyku Python
- Zkompilovanou a přeloženou aplikaci pro vytváření textových přepisů ve formátu EXE včetně nástroje *pdftotext*